UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/003,683	11/02/2001	Theodore S. Webb	SURG:162	4096

7590 05/16/2005	EXAMINER
O'KEEFE, EGAN & PETERMAN, L.L.P.	AVELLINO, JOSEPH E
Building C, Suite 200	

ART UNIT	PAPER NUMBER
2143	

1101 Capital of Texas Highway South
Austin, TX 78746

DATE MAILED: 05/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	**Applicant(s)**
Office Action Summary	10/003,683	WEBB ET AL.
	Examiner	**Art Unit**
	Joseph E. Avellino	2143

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>1</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>02 November 2001</u>.

2a)☐ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-154</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) <u>1-154</u> are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1.   Restriction to one of the following inventions is required under 35 U.S.C. 121:

I.   Claims 1-28, drawn to a system for loading an executable image onto at least one image receiver, said system comprising at least one image source, an image receiver capable of communicating across said distributed interconnect for loading on to said at least one image receiver across said distributed interconnect for loading onto said at least one image receiver, wherein the image comprises a diagnostic image, an initial image, boot code, OS, API, an application, wherein said image source comprises a management processing engine, and wherein said image receiver comprises an application processing engine, wherein the information management system comprises multiple image sources, multiple image receivers, wherein the image source has access to a plurality of different executable images, further comprising selecting and communicating a second executable image, wherein the image remains quiescent until an execution signal is sent from said image sender, classified in class 709, subclass 220.

II.   Claims 29-96, drawn to a system for interfacing a first processing object with a second processing object, comprising a first processing engine with a first processing object residing thereon, a second processing engine

coupled to said first processing engine by a distributed interconnect

having said second processing object residing thereon, capable of

interfacing across said distributed interconnect, wherein said first

processing engine comprises an application processing engine and the

second processing engine comprises a storage processing engine,

wherein the first processing object comprises an application object,

wherein the second processing object comprises a buffer/cache object

specific to application object, further comprising at least two first

processing engines having at least one respective first processing object

residing thereon, a second processing engine with a second processing

object residing thereon, wherein the characteristics of a given second

processing object residing on at least one of said second processing

engines differs from the characteristics of an other second processing

object residing on another one of said second processing engines,

wherein each of the first processing engines comprises a first application

processing engine of a content delivery system, and wherein each of said

second processing engines comprises a storage processing engine of

said content delivery system, to allow a first processing engine on which

said first processing engine resides to retrieve content from a content

source using a second processing engine on which said selected second

processing object resides, classified in class 709, subclass 217.

III.   Claims 97-119, drawn to a method of coordinating a group of multiple

processing engines in the performance of an operating task comprising

broadcasting a multicast message to said group of multiple processing

engines across a distributed interconnect, said multicast facilitating the

performance of said operating task, wherein said operating task comprises

a failover operation, a load-balancing operation, a debugging operation,

an operation to monitor a status of one or more information management

resources, broadcasting said multicast message across said distributed

interconnect to keep one or more of said group of processing engines

apprised of the status of one or more individual members of said group,

implementing said failover operation upon absence of said period

multicast communications from a failed processing engine by using

another processing engine to assume the load or tasks of said failed

processing engine and broadcasting a multicast failure alarm from a failed

processing engine of said group of engines to other members of said

group and implementing said failover operation upon broadcast of said

failure alarm to assume the load or tasks of said failed processing engine,

using one or more designated members of said group to monitor and

detect failures of one or more other members of the group, and broadcast

a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast

communications to keep members apprised of the load of the engine,

monitor and detect workload level, upon detection of an imbalance,

broadcast to other members to implement a load balancing operation to

transfer workload among the other members, allowing the multicast

message accessible for debug analysis by a human operator, classified in

class 709, subclass 224.

IV.   Claims 120-130, drawn to a method of analyzing software code running on

first processing engine, communicating debug information associated with

said code from said first processing engine to a second processing engine

across a distributed interconnect, viewing, analyzing or storing said debug

information on said processing engine, analyzing said debug information

using said second processing engine, classified in class 717, subclass

124.

V.   Claims 131-154, drawn to a method of managing the manipulation of

information among a group of multiple processing engines in an

information management environment each engine capable of performing

one or more information manipulation tasks, receiving first and second

request for information management, selecting a processing flow path

among a group of processing engines to perform a first selected

combination of information manipulation tasks associated with said first

request for information management, a second processing flow path in

order to perform a second selected combination of information

manipulation tasks associated with said second request for information

management, wherein the engines are couple together by a distributed interconnect, wherein the first and second processing flows are different each selected using a distributed interconnect, each of said multiple processing engines is assigned separate information manipulation tasks in an asymmetrical multi-processor configuration, selecting the processing flow paths is based on a first and second parameters associated with each of said first and second requests for information management, whereby each request for information management is associated with a particular user or class of users generating each of said first and second requests for information management, wherein the first and second parameters are priority-indicative parameters, wherein one or more information processing engines are capable of performing one or more of the same core information manipulation tasks as performed by one or more processing engines of said processing flow path, wherein the selectable information manipulation tasks comprise data encryption, data compression, security function, transcoding, content transformation, filtering based on metadata, metadata transformation, altering at least a portion of a processing flow path based upon said recognized parameter, classified in class 713, subclass 201.

The inventions are distinct, each from the other because of the following reasons:

2.   Inventions I-V are related as subcombinations disclosed as usable together in a

single combination. The subcombinations are distinct from each other if they are shown

to be separately usable. In the instant case, invention I has separate utility such as a

system for loading an executable image onto at least one image receiver, said system

comprising at least one image source, an image receiver capable of communicating

across said distributed interconnect for loading on to said at least one image receiver

across said distributed interconnect for loading onto said at least one image receiver,

wherein the image comprises a diagnostic image, an initial image, boot code, OS, API,

an application, wherein said image source comprises a management processing

engine, and wherein said image receiver comprises an application processing engine,

wherein the information management system comprises multiple image sources,

multiple image receivers, wherein the image source has access to a plurality of different

executable images, further comprising selecting and communicating a second

executable image, wherein the image remains quiescent until an execution signal is

sent from said image sender and lacking a system for interfacing a first processing

object with a second processing object, comprising a first processing engine with a first

processing object residing thereon, a second processing engine coupled to said first

processing engine by a distributed interconnect having said second processing object

residing thereon, capable of interfacing across said distributed interconnect, wherein

said first processing engine comprises an application processing engine and the second

processing engine comprises a storage processing engine, wherein the first processing

object comprises an application object, wherein the second processing object

comprises a buffer/cache object specific to application object, further comprising at least

two first processing engines having at least one respective first processing object

residing thereon, a second processing engine with a second processing object residing

thereon, wherein the characteristics of a given second processing object residing on at

least one of said second processing engines differs from the characteristics of an other

second processing object residing on another one of said second processing engines,

wherein each of the first processing engines comprises a first application processing

engine of a content delivery system, and wherein each of said second processing

engines comprises a storage processing engine of said content delivery system, to

allow a first processing engine on which said first processing engine resides to retrieve

content from a content source using a second processing engine on which said selected

second processing object resides, drawn to a method of coordinating a group of multiple

processing engines in the performance of an operating task comprising broadcasting a

multicast message to said group of multiple processing engines across a distributed

interconnect, said multicast facilitating the performance of said operating task, wherein

said operating task comprises a failover operation, a load-balancing operation, a

debugging operation, an operation to monitor a status of one or more information

management resources, broadcasting said multicast message across said distributed

interconnect to keep one or more of said group of processing engines apprised of the

status of one or more individual members of said group, implementing said failover

operation upon absence of said period multicast communications from a failed

processing engine by using another processing engine to assume the load or tasks of

said failed processing engine and broadcasting a multicast failure alarm from a failed

processing engine of said group of engines to other members of said group and

implementing said failover operation upon broadcast of said failure alarm to assume the

load or tasks of said failed processing engine, using one or more designated members

of said group to monitor and detect failures of one or more other members of the group,

and broadcast a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast communications to

keep members apprised of the load of the engine, monitor and detect workload level,

upon detection of an imbalance, broadcast to other members to implement a load

balancing operation to transfer workload among the other members, allowing the

multicast message accessible for debug analysis by a human operator, a method of

analyzing software code running on first processing engine, communicating debug

information associated with said code from said first processing engine to a second

processing engine across a distributed interconnect, viewing, analyzing or storing said

debug information on said processing engine, analyzing said debug information using

said second processing engine, and a method of managing the manipulation of

information among a group of multiple processing engines in an information

management environment each engine capable of performing one or more information

manipulation tasks, receiving first and second request for information management,

selecting a processing flow path among a group of processing engines to perform a first

selected combination of information manipulation tasks associated with said first

request for information management, a second processing flow path in order to perform

a second selected combination of information manipulation tasks associated with said

second request for information management, wherein the engines are couple together

by a distributed interconnect, wherein the first and second processing flows are different

each selected using a distributed interconnect, each of said multiple processing engines

is assigned separate information manipulation tasks in an asymmetrical multi-processor

configuration, selecting the processing flow paths is based on a first and second

parameters associated with each of said first and second requests for information

management, whereby each request for information management is associated with a

particular user or class of users generating each of said first and second requests for

information management, wherein the first and second parameters are priority-indicative

parameters, wherein one or more information processing engines are capable of

performing one or more of the same core information manipulation tasks as performed

by one or more processing engines of said processing flow path, wherein the selectable

information manipulation tasks comprise data encryption, data compression, security

function, transcoding, content transformation, filtering based on metadata, metadata

transformation, altering at least a portion of a processing flow path based upon said

recognized parameter. Invention II has separate utility such as a system for interfacing

a first processing object with a second processing object, comprising a first processing

engine with a first processing object residing thereon, a second processing engine

coupled to said first processing engine by a distributed interconnect having said second

processing object residing thereon, capable of interfacing across said distributed

interconnect, wherein said first processing engine comprises an application processing

engine and the second processing engine comprises a storage processing engine,

wherein the first processing object comprises an application object, wherein the second

processing object comprises a buffer/cache object specific to application object, further

comprising at least two first processing engines having at least one respective first

processing object residing thereon, a second processing engine with a second

processing object residing thereon, wherein the characteristics of a given second

processing object residing on at least one of said second processing engines differs

from the characteristics of an other second processing object residing on another one of

said second processing engines, wherein each of the first processing engines

comprises a first application processing engine of a content delivery system, and

wherein each of said second processing engines comprises a storage processing

engine of said content delivery system, to allow a first processing engine on which said

first processing engine resides to retrieve content from a content source using a second

processing engine on which said selected second processing object resides, and

lacking a system for loading an executable image onto at least one image receiver, said

system comprising at least one image source, an image receiver capable of

communicating across said distributed interconnect for loading on to said at least one

image receiver across said distributed interconnect for loading onto said at least one

image receiver, wherein the image comprises a diagnostic image, an initial image, boot

code, OS, API, an application, wherein said image source comprises a management

processing engine, and wherein said image receiver comprises an application

processing engine, wherein the information management system comprises multiple

image sources, multiple image receivers, wherein the image source has access to a

plurality of different executable images, further comprising selecting and communicating

a second executable image, wherein the image remains quiescent until an execution

signal is sent from said image sender, a method of coordinating a group of multiple

processing engines in the performance of an operating task comprising broadcasting a

multicast message to said group of multiple processing engines across a distributed

interconnect, said multicast facilitating the performance of said operating task, wherein

said operating task comprises a failover operation, a load-balancing operation, a

debugging operation, an operation to monitor a status of one or more information

management resources, broadcasting said multicast message across said distributed

interconnect to keep one or more of said group of processing engines apprised of the

status of one or more individual members of said group, implementing said failover

operation upon absence of said period multicast communications from a failed

processing engine by using another processing engine to assume the load or tasks of

said failed processing engine and broadcasting a multicast failure alarm from a failed

processing engine of said group of engines to other members of said group and

implementing said failover operation upon broadcast of said failure alarm to assume the

load or tasks of said failed processing engine, using one or more designated members

of said group to monitor and detect failures of one or more other members of the group,

and broadcast a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast communications to

keep members apprised of the load of the engine, monitor and detect workload level,

upon detection of an imbalance, broadcast to other members to implement a load

balancing operation to transfer workload among the other members, allowing the

multicast message accessible for debug analysis by a human operator, a method of

analyzing software code running on first processing engine, communicating debug

information associated with said code from said first processing engine to a second

processing engine across a distributed interconnect, viewing, analyzing or storing said

debug information on said processing engine, analyzing said debug information using

said second processing engine, and a method of managing the manipulation of

information among a group of multiple processing engines in an information

management environment each engine capable of performing one or more information

manipulation tasks, receiving first and second request for information management,

selecting a processing flow path among a group of processing engines to perform a first

selected combination of information manipulation tasks associated with said first

request for information management, a second processing flow path in order to perform

a second selected combination of information manipulation tasks associated with said

second request for information management, wherein the engines are couple together

by a distributed interconnect, wherein the first and second processing flows are different

each selected using a distributed interconnect, each of said multiple processing engines

is assigned separate information manipulation tasks in an asymmetrical multi-processor

configuration, selecting the processing flow paths is based on a first and second

parameters associated with each of said first and second requests for information

management, whereby each request for information management is associated with a

particular user or class of users generating each of said first and second requests for

information management, wherein the first and second parameters are priority-indicative

parameters, wherein one or more information processing engines are capable of

performing one or more of the same core information manipulation tasks as performed

by one or more processing engines of said processing flow path, wherein the selectable

information manipulation tasks comprise data encryption, data compression, security

function, transcoding, content transformation, filtering based on metadata, metadata

transformation, altering at least a portion of a processing flow path based upon said

recognized parameter. Invention III has separate utility such as a method of

coordinating a group of multiple processing engines in the performance of an operating

task comprising broadcasting a multicast message to said group of multiple processing

engines across a distributed interconnect, said multicast facilitating the performance of

said operating task, wherein said operating task comprises a failover operation, a load-

balancing operation, a debugging operation, an operation to monitor a status of one or

more information management resources, broadcasting said multicast message across

said distributed interconnect to keep one or more of said group of processing engines

apprised of the status of one or more individual members of said group, implementing

said failover operation upon absence of said period multicast communications from a

failed processing engine by using another processing engine to assume the load or

tasks of said failed processing engine and broadcasting a multicast failure alarm from a

failed processing engine of said group of engines to other members of said group and

implementing said failover operation upon broadcast of said failure alarm to assume the

load or tasks of said failed processing engine, using one or more designated members

of said group to monitor and detect failures of one or more other members of the group,

and broadcast a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast communications to

keep members apprised of the load of the engine, monitor and detect workload level,

upon detection of an imbalance, broadcast to other members to implement a load

balancing operation to transfer workload among the other members, allowing the

multicast message accessible for debug analysis by a human operator, and lacking a

system for loading an executable image onto at least one image receiver, said system

comprising at least one image source, an image receiver capable of communicating

across said distributed interconnect for loading on to said at least one image receiver

across said distributed interconnect for loading onto said at least one image receiver,

wherein the image comprises a diagnostic image, an initial image, boot code, OS, API,

an application, wherein said image source comprises a management processing

engine, and wherein said image receiver comprises an application processing engine,

wherein the information management system comprises multiple image sources,

multiple image receivers, wherein the image source has access to a plurality of different

executable images, further comprising selecting and communicating a second

executable image, wherein the image remains quiescent until an execution signal is

sent from said image sender, a system for interfacing a first processing object with a

second processing object, comprising a first processing engine with a first processing

object residing thereon, a second processing engine coupled to said first processing

engine by a distributed interconnect having said second processing object residing

thereon, capable of interfacing across said distributed interconnect, wherein said first

processing engine comprises an application processing engine and the second

processing engine comprises a storage processing engine, wherein the first processing

object comprises an application object, wherein the second processing object

comprises a buffer/cache object specific to application object, further comprising at least

two first processing engines having at least one respective first processing object

residing thereon, a second processing engine with a second processing object residing

thereon, wherein the characteristics of a given second processing object residing on at

least one of said second processing engines differs from the characteristics of an other

second processing object residing on another one of said second processing engines,

wherein each of the first processing engines comprises a first application processing

engine of a content delivery system, and wherein each of said second processing

engines comprises a storage processing engine of said content delivery system, to

allow a first processing engine on which said first processing engine resides to retrieve

content from a content source using a second processing engine on which said selected

second processing object resides, a method of analyzing software code running on first

processing engine, communicating debug information associated with said code from

said first processing engine to a second processing engine across a distributed

interconnect, viewing, analyzing or storing said debug information on said processing

engine, analyzing said debug information using said second processing engine, and a

method of managing the manipulation of information among a group of multiple

processing engines in an information management environment each engine capable of

performing one or more information manipulation tasks, receiving first and second

request for information management, selecting a processing flow path among a group of

processing engines to perform a first selected combination of information manipulation

tasks associated with said first request for information management, a second

processing flow path in order to perform a second selected combination of information

manipulation tasks associated with said second request for information management,

wherein the engines are couple together by a distributed interconnect, wherein the first

and second processing flows are different each selected using a distributed

interconnect, each of said multiple processing engines is assigned separate information

manipulation tasks in an asymmetrical multi-processor configuration, selecting the

processing flow paths is based on a first and second parameters associated with each

of said first and second requests for information management, whereby each request

for information management is associated with a particular user or class of users

generating each of said first and second requests for information management, wherein

the first and second parameters are priority-indicative parameters, wherein one or more

information processing engines are capable of performing one or more of the same core

information manipulation tasks as performed by one or more processing engines of said

processing flow path, wherein the selectable information manipulation tasks comprise

data encryption, data compression, security function, transcoding, content

transformation, filtering based on metadata, metadata transformation, altering at least a

portion of a processing flow path based upon said recognized parameter. Invention IV

has separate utility such as a method of analyzing software code running on first

processing engine, communicating debug information associated with said code from

said first processing engine to a second processing engine across a distributed

interconnect, viewing, analyzing or storing said debug information on said processing

engine, analyzing said debug information using said second processing engine, and

lacks a system for loading an executable image onto at least one image receiver, said

system comprising at least one image source, an image receiver capable of

communicating across said distributed interconnect for loading on to said at least one

image receiver across said distributed interconnect for loading onto said at least one

image receiver, wherein the image comprises a diagnostic image, an initial image, boot

code, OS, API, an application, wherein said image source comprises a management

processing engine, and wherein said image receiver comprises an application

processing engine, wherein the information management system comprises multiple

image sources, multiple image receivers, wherein the image source has access to a

plurality of different executable images, further comprising selecting and communicating

a second executable image, wherein the image remains quiescent until an execution

signal is sent from said image sender, a system for interfacing a first processing object

with a second processing object, comprising a first processing engine with a first

processing object residing thereon, a second processing engine coupled to said first

processing engine by a distributed interconnect having said second processing object

residing thereon, capable of interfacing across said distributed interconnect, wherein

said first processing engine comprises an application processing engine and the second

processing engine comprises a storage processing engine, wherein the first processing

object comprises an application object, wherein the second processing object

comprises a buffer/cache object specific to application object, further comprising at least

two first processing engines having at least one respective first processing object

residing thereon, a second processing engine with a second processing object residing

thereon, wherein the characteristics of a given second processing object residing on at

least one of said second processing engines differs from the characteristics of an other

second processing object residing on another one of said second processing engines,

wherein each of the first processing engines comprises a first application processing

engine of a content delivery system, and wherein each of said second processing

engines comprises a storage processing engine of said content delivery system, to

allow a first processing engine on which said first processing engine resides to retrieve

content from a content source using a second processing engine on which said selected

second processing object resides, a method of coordinating a group of multiple

processing engines in the performance of an operating task comprising broadcasting a

multicast message to said group of multiple processing engines across a distributed

interconnect, said multicast facilitating the performance of said operating task, wherein

said operating task comprises a failover operation, a load-balancing operation, a

debugging operation, an operation to monitor a status of one or more information

management resources, broadcasting said multicast message across said distributed

interconnect to keep one or more of said group of processing engines apprised of the

status of one or more individual members of said group, implementing said failover

operation upon absence of said period multicast communications from a failed

processing engine by using another processing engine to assume the load or tasks of

said failed processing engine and broadcasting a multicast failure alarm from a failed

processing engine of said group of engines to other members of said group and

implementing said failover operation upon broadcast of said failure alarm to assume the

load or tasks of said failed processing engine, using one or more designated members

of said group to monitor and detect failures of one or more other members of the group,

and broadcast a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast communications to

keep members apprised of the load of the engine, monitor and detect workload level,

upon detection of an imbalance, broadcast to other members to implement a load

balancing operation to transfer workload among the other members, allowing the

multicast message accessible for debug analysis by a human operator, and a method of

managing the manipulation of information among a group of multiple processing

engines in an information management environment each engine capable of performing

one or more information manipulation tasks, receiving first and second request for

information management, selecting a processing flow path among a group of

processing engines to perform a first selected combination of information manipulation

tasks associated with said first request for information management, a second

processing flow path in order to perform a second selected combination of information

manipulation tasks associated with said second request for information management,

wherein the engines are couple together by a distributed interconnect, wherein the first

and second processing flows are different each selected using a distributed

interconnect, each of said multiple processing engines is assigned separate information

manipulation tasks in an asymmetrical multi-processor configuration, selecting the

processing flow paths is based on a first and second parameters associated with each

of said first and second requests for information management, whereby each request

for information management is associated with a particular user or class of users

generating each of said first and second requests for information management, wherein

the first and second parameters are priority-indicative parameters, wherein one or more

information processing engines are capable of performing one or more of the same core

information manipulation tasks as performed by one or more processing engines of said

processing flow path, wherein the selectable information manipulation tasks comprise

data encryption, data compression, security function, transcoding, content

transformation, filtering based on metadata, metadata transformation, altering at least a

portion of a processing flow path based upon said recognized parameter. Invention V

has separate utility such as a method of managing the manipulation of information

among a group of multiple processing engines in an information management

environment each engine capable of performing one or more information manipulation

tasks, receiving first and second request for information management, selecting a

processing flow path among a group of processing engines to perform a first selected

combination of information manipulation tasks associated with said first request for

information management, a second processing flow path in order to perform a second

selected combination of information manipulation tasks associated with said second

request for information management, wherein the engines are couple together by a

distributed interconnect, wherein the first and second processing flows are different

each selected using a distributed interconnect, each of said multiple processing engines

is assigned separate information manipulation tasks in an asymmetrical multi-processor

configuration, selecting the processing flow paths is based on a first and second

parameters associated with each of said first and second requests for information

management, whereby each request for information management is associated with a

particular user or class of users generating each of said first and second requests for

information management, wherein the first and second parameters are priority-indicative

parameters, wherein one or more information processing engines are capable of

performing one or more of the same core information manipulation tasks as performed

by one or more processing engines of said processing flow path, wherein the selectable

information manipulation tasks comprise data encryption, data compression, security

function, transcoding, content transformation, filtering based on metadata, metadata

transformation, altering at least a portion of a processing flow path based upon said

recognized parameter, and lacks a system for loading an executable image onto at least

one image receiver, said system comprising at least one image source, an image

receiver capable of communicating across said distributed interconnect for loading on to

said at least one image receiver across said distributed interconnect for loading onto

said at least one image receiver, wherein the image comprises a diagnostic image, an

initial image, boot code, OS, API, an application, wherein said image source comprises

a management processing engine, and wherein said image receiver comprises an

application processing engine, wherein the information management system comprises

multiple image sources, multiple image receivers, wherein the image source has access

to a plurality of different executable images, further comprising selecting and

communicating a second executable image, wherein the image remains quiescent until

an execution signal is sent from said image sender, a system for interfacing a first

processing object with a second processing object, comprising a first processing engine

with a first processing object residing thereon, a second processing engine coupled to

said first processing engine by a distributed interconnect having said second processing

object residing thereon, capable of interfacing across said distributed interconnect,

wherein said first processing engine comprises an application processing engine and

the second processing engine comprises a storage processing engine, wherein the first

processing object comprises an application object, wherein the second processing

object comprises a buffer/cache object specific to application object, further comprising

at least two first processing engines having at least one respective first processing

object residing thereon, a second processing engine with a second processing object

residing thereon, wherein the characteristics of a given second processing object

residing on at least one of said second processing engines differs from the

characteristics of an other second processing object residing on another one of said

second processing engines, wherein each of the first processing engines comprises a

first application processing engine of a content delivery system, and wherein each of

said second processing engines comprises a storage processing engine of said content

delivery system, to allow a first processing engine on which said first processing engine

resides to retrieve content from a content source using a second processing engine on

which said selected second processing object resides, a method of coordinating a group

of multiple processing engines in the performance of an operating task comprising

broadcasting a multicast message to said group of multiple processing engines across a

distributed interconnect, said multicast facilitating the performance of said operating

task, wherein said operating task comprises a failover operation, a load-balancing

operation, a debugging operation, an operation to monitor a status of one or more

information management resources, broadcasting said multicast message across said

distributed interconnect to keep one or more of said group of processing engines

apprised of the status of one or more individual members of said group, implementing

said failover operation upon absence of said period multicast communications from a

failed processing engine by using another processing engine to assume the load or

.tasks of said failed processing engine and broadcasting a multicast failure alarm from a

failed processing engine of said group of engines to other members of said group and

implementing said failover operation upon broadcast of said failure alarm to assume the

load or tasks of said failed processing engine, using one or more designated members

of said group to monitor and detect failures of one or more other members of the group,

and broadcast a multicast failure alarm to other members of the group and assume the

load or tasks of the failed processing engine, broadcast multicast communications to

keep members apprised of the load of the engine, monitor and detect workload level,

upon detection of an imbalance, broadcast to other members to implement a load

balancing operation to transfer workload among the other members, allowing the

multicast message accessible for debug analysis by a human operator, and a method of

analyzing software code running on first processing engine, communicating debug

information associated with said code from said first processing engine to a second

processing engine across a distributed interconnect, viewing, analyzing or storing said

debug information on said processing engine, analyzing said debug information using

said second processing engine. See MPEP § 806.05(d).

3.   Because these inventions are distinct for the reasons given above and have

acquired a separate status in the art as shown by their different classification, restriction

for examination purposes as indicated is proper.

Because these inventions are distinct for the reasons given above and the

search required for each group is different as shown below, restriction for examination

purposes as indicated is proper.

Group I requires a search of 709/220, which is not required for Groups II-V.

Group II requires a search of 709/219, which is not required for Groups I, III-V.

Group III requires a search of 709/224, which is not required for Groups I-II, IV-V.

Group IV requires a search of 717/124, which is not required for Groups I-III, V.

Group V requires a search of 713/201, which is not required for Groups I-IV.

4.   Applicant is reminded that upon the cancellation of claims to a non-elected

invention, the inventorship must be amended in compliance with 37 CFR 1.48(b) if one

or more of the currently named inventors is no longer an inventor of at least one claim

remaining in the application. Any amendment of inventorship must be accompanied by a request under 37 CFR 1.48(b) and by the fee required under 37 CFR 1.17(i).
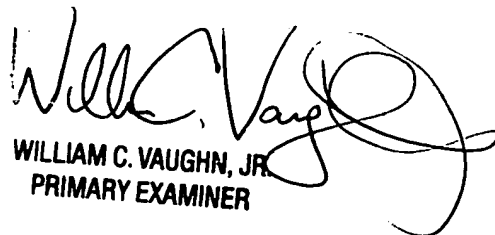
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph E. Avellino whose telephone number is (571) 272-3905. The examiner can normally be reached on Monday-Friday 7:00-4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David A. Wiley can be reached on (571) 272-3923. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JEA
May 10, 2005

WILLIAM C. VAUGHN, JR.
PRIMARY EXAMINER